

[0034] As clearly shown in FIG. 2(A) and (B), in the conventional program written in VLIW, when there are two or more consecutive instructions of “NOP” in one instruction to a CPU, a program written in VLIW according to the present invention can be obtained by replacing the first instruction of “NOP” with an instruction of “STOP [CPU name]” and the last instruction of “NOP” with an instruction of “RET [CPU name]”.

[0044] The clock control unit 12 determines whether or not the readout instruction of α bytes includes “STOP [CPU name]” or “RET [CPU name]”. The clock control unit 12 then stops, when “STOP [CPU name]” is included, supplying a clock signal to the CPU which corresponds to the [CPU name], and starts (restarts), when “RET [CPU name]” is included, supplying the clock signal to the CPU which corresponds to the [CPU name].

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-123331

(43)Date of publication of application : 26.04.2002

(51)Int.Cl.

G06F 1/04

G06F 1/32

(21)Application number : 2000-316969

(71)Applicant : FUJITSU LTD

(22)Date of filing : 17.10.2000

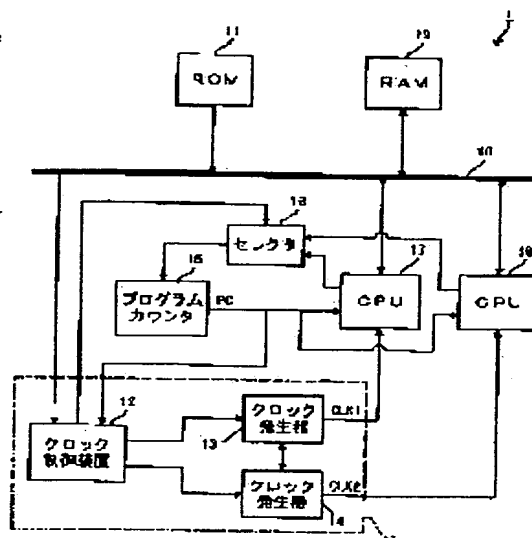
(72)Inventor : MATSUI SATOSHI
IINO HIDEYUKI

(54) INFORMATION PROCESSING SYSTEM, AND UNIT AND METHOD FOR CLOCK CONTROL

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the power consumption of an information processing system equipped with multiple processors.

SOLUTION: On a ROM 11, a program including instructions for stopping the supply of clock signals from a clock generator 13 to a CPU 17 or from a clock generator 14 to a CPU 18 in addition to instructions that the CPU 17 or 18 processes is stored. The clock control unit 12 reads an instruction of the program out of the ROM 11 and stops supplying the clock signal from the clock generator 13 when the instruction is an instruction for stopping the supply of the clock signal to the CPU 17 or stops supplying the clock signal from the clock generator 14 when the instruction is an instruction for stopping the supply of the clock signal to the CPU 18.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-123331

(P2002-123331A)

(43) 公開日 平成14年4月26日 (2002. 4. 26)

(51) Int.Cl.⁷

識別記号

F I

データ* (参考)

G 0 6 F 1/04
1/32

3 0 1

C 0 6 F 1/04
1/00

3 0 1 C 5 B 0 1 1
3 3 2 Z 5 B 0 7 9

審査請求 未請求 請求項の数 7 O L (全 10 頁)

(21) 出願番号 特願2000-316969 (P2000-316969)

(22) 出願日 平成12年10月17日 (2000. 10. 17)

(71) 出願人 000003273

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 松井 聡

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 飯野 秀之

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 100094525

弁理士 土井 健二 (外1名)

Fターム(参考) 5B011 LL13

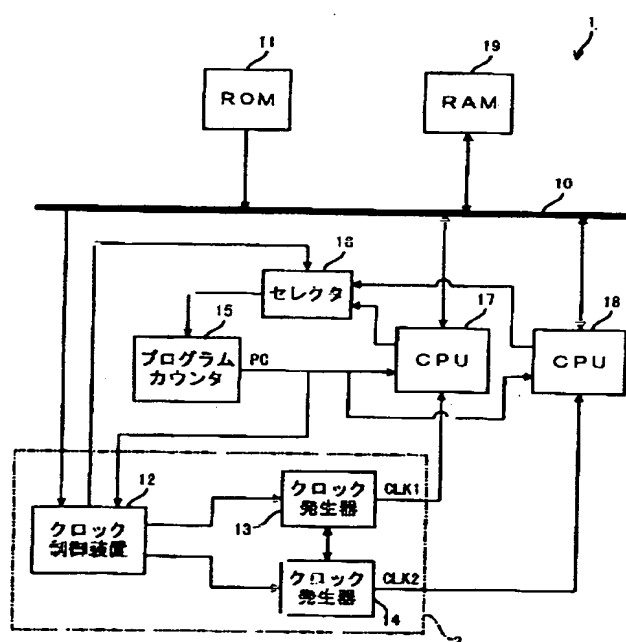
5B079 AA06 BA01 BA12 BB01 BC01

(54) 【発明の名称】 情報処理システム、クロック制御装置、およびクロック制御方法

(57) 【要約】

【課題】 複数の処理装置を備えている情報処理システムにおいて、消費電力を軽減する。

【解決手段】 ROM 11 には、CPU 17 または 18 が処理する命令に加えて、クロック発生器 13 から CPU 17 への、または、クロック発生器 14 から CPU 18 への各クロック信号の供給を停止する命令を含んだプログラムが記憶されている。クロック制御装置 12 は、ROM 11 からプログラムの命令を読み出し、この命令が CPU 17 へのクロック信号の供給を停止する命令である場合には、クロック発生器 13 からのクロック信号の供給を停止し、この命令が CPU 18 へのクロック信号の供給を停止する命令である場合には、クロック発生器 14 からのクロック信号の供給を停止する。



【特許請求の範囲】

【請求項1】 クロック発生器と、該クロック発生器が生成するクロック信号がそれぞれ供給され、該クロック信号に同期してそれぞれに割り当てられた処理を実行する複数の処理装置とを備えている情報処理システムにおいて、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給するクロック制御装置を備えている、ことを特徴とする情報処理システム。

【請求項2】 請求項1において、前記クロック制御装置が、前記クロック信号の供給が停止された処理装置が処理を再開する場合には、該処理装置へのクロック信号の供給を開始し、前記処理の実行時の周期よりも長い周期のクロック信号を供給された処理装置が処理を再開する場合には、該処理装置へのクロック信号の周期を元に戻す、ことを特徴とする情報処理システム。

【請求項3】 請求項1において、プログラムを記憶するメモリと、前記メモリに記憶された前記プログラムを構成する命令のうち実行対象となる命令が記憶されたアドレスを順次示すプログラム・カウンタを生成して、前記複数の処理装置の少なくとも2つの処理装置に与えるプログラム・カウンタと、をさらに備え、前記少なくとも2つの処理装置が、前記メモリに記憶されたプログラムを前記プログラム・カウンタから与えられる共通のプログラム・カウンタに従ってそれぞれ実行する、ことを特徴とする情報処理システム。

【請求項4】 請求項3において、前記プログラムが、前記クロック信号の前記処理装置への供給を停止する停止命令と、前記停止命令の対象となる処理装置を指定する指定情報とを備え、前記クロック制御装置が、前記停止命令および前記指定情報に従って、前記指定情報により指定された処理装置へのクロック信号の供給を停止する、ことを特徴とする情報処理システム。

【請求項5】 請求項2において、プログラムを記憶するメモリと、前記メモリに記憶された前記プログラムを構成する命令のうち実行対象となる命令が記憶されたアドレスを順次示すプログラム・カウンタを生成して、前記複数の処理装置の少なくとも2つの処理装置に与えるプログラム・カウンタと、をさらに備え、前記少なくとも2つの処理装置が、前記メモリに記憶されたプログラムを前記プログラム・カウンタから与えられる共通のプログラム・カウンタに従ってそれぞれ実行し、前記プログラムが、前記クロック信号の前記処理装置への供給を開始する開始命令と、前記開始命令の対象となる処理装置を指定する指定情報とを備え、前記クロック制御装置が、前記開始命令および前記指定情報に従って、前記指定情報により指定された処理装置へのクロック信号の供給を開始する、ことを特徴とする情報処理システム。

【請求項6】 クロック信号の供給を受け、該クロック

信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御装置であって、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給するようにクロック信号の供給を制御する、クロック制御装置。

【請求項7】 クロック信号の供給を受け、該クロック信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御方法であって、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給する、クロック制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理システムに関し、特に、クロック発生器と、該クロック発生器が生成するクロック信号がそれぞれ供給され、該クロック信号に同期してそれぞれに割り当てられた処理を実行する複数の処理装置とを備えている情報処理システムに関する。

【0002】また、本発明は、クロック信号の供給を受け、該クロック信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御装置およびその方法に関する。

【0003】

【従来の技術】コンピュータ等の情報処理システムの処理の高速化を図るために、1つの情報処理システム内に複数の処理装置（演算装置、中央処理装置（CPU）等）を設け、これらの処理装置間で負荷分散処理、機能分散処理等が行われている。

【0004】これらの複数の処理装置は、クロック発生器が供給するクロック信号に同期して処理を行っている。

【0005】

【発明が解決しようとする課題】しかし、このような複数の処理装置を備えている情報処理システムでは、複数の処理装置のすべてが常に処理を行っているわけではなく、ある処理装置は処理を行っているが、他の処理装置は処理を行っていない場合がある。このような場合であっても、従来の情報処理システムでは、すべての処理装置に、クロック信号が供給されていた。

【0006】このため、処理を行っていない処理装置も、クロック信号が供給されることから、処理を行っている処理装置と同様に電力を消費していた。また、電力消費に伴う発熱の増大、および、この熱を外部に放出するためのファン等による電力消費の増大も問題となっていた。

【0007】一方、このような情報処理システムには、

複数の処理装置がプログラムを実行する場合に、各処理装置が個別のプログラム・カウンタを備え、自己の備えたプログラム・カウンタに従ってプログラムを実行するものがある。

【0008】しかし、個別のプログラム・カウンタに従って処理を行うと、各処理装置によってプログラム・カウンタのカウント値が相違するので、各処理装置が実行しているプログラムの部分が異なり、このため、プログラムのデバッグや情報処理システムを構成するハードウェアのデバッグが困難となる場合があった。

【0009】本発明は、このような状況に鑑みなされたものであり、その第1の目的は、複数の処理装置を備えている情報処理システムにおいて、消費電力を軽減することにある。

【0010】また、本発明の第2の目的は、複数の処理装置を備えている情報処理システムにおいて、消費電力を軽減しつつ、かつ、プログラムのデバッグをも行い易くすることにある。

【0011】

【課題を解決するための手段】前記第1の目的を達成するために、本発明に係る情報処理システムは、クロック発生器と、該クロック発生器が生成するクロック信号がそれぞれ供給され、該クロック信号に同期してそれぞれに割り当てられた処理を実行する複数の処理装置とを備えている情報処理システムにおいて、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給するクロック制御装置を備えている、ことを特徴とする。

【0012】また、本発明に係るクロック制御装置は、クロック信号の供給を受け、該クロック信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御装置であって、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給するようにクロック信号の供給を制御する。

【0013】さらに、本発明に係るクロック制御方法は、クロック信号の供給を受け、該クロック信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御方法であって、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給する。

【0014】本発明によると、複数の処理装置のうち、処理を行っていない処理装置へのクロック信号の供給が停止される。これにより、クロック信号の供給が停止された処理装置の電力消費が軽減されるので、情報処理システム全体の消費電力が軽減される。また、消費電力の軽減は、発熱量の軽減にもつながり、発熱の増大、およ

び、この熱を外部に放出するためのファン等による電力消費の増大も防止できる。その結果、前記第1の目的が達成される。

【0015】好ましくは、前記クロック制御装置は、前記クロック信号の供給が停止された処理装置が処理を再開する場合には、該処理装置へのクロック信号の供給を開始し、前記処理の実行時の周期よりも長い周期のクロック信号を供給された処理装置が処理を再開する場合には、該処理装置へのクロック信号の周期を元に戻す。

【0016】本発明によると、処理装置が処理を再開する場合には、クロック信号の供給が再開され、または、クロック信号の周期が元に戻されるので、その処理装置は該処理装置が有する処理性能によって通常通り処理を実行することができる。

【0017】前記第2の目的を達成するために、本発明の一実施の形態においては、前記情報処理システムが、プログラムを記憶するメモリと、前記メモリに記憶された前記プログラムを構成する命令のうち実行対象となる命令が記憶されたアドレスを順次示すプログラム・カウンタを生成して、前記複数の処理装置の少なくとも2つの処理装置に与えるプログラム・カウンタと、をさらに備え、前記少なくとも2つの処理装置が、前記メモリに記憶されたプログラムを前記プログラム・カウンタから与えられる共通のプログラム・カウンタに従ってそれぞれ実行する。

【0018】本発明の一実施の形態によると、複数の処理装置のうち、処理を行っていない処理装置へのクロック信号の供給が停止されるとともに、複数の処理装置は、共通のプログラム・カウンタに基づいてプログラムを実行する。その結果、処理を行っていない処理装置へのクロック信号の供給を停止することにより、消費電力の軽減が図られ、また、複数の処理装置が共通のプログラム・カウンタに基づいてプログラムを実行するので、プログラムのデバッグおよび情報処理システムを構成するハードウェアのデバッグが容易となる。これにより、前記第2の目的が達成される。

【0019】

【発明の実施の形態】<第1の実施の形態>図1は、本発明の第1の実施の形態に係る情報処理システム1の構成を示すブロック図である。この情報処理システム1は、システム・バス10、ROM11、クロック制御装置12、クロック発生器13および14、プログラム・カウンタ15、セクタ16、RAM19、ならびに「処理装置」の一例としてのCPU17および18を備えている。符号3は、クロック制御装置12ならびにクロック発生器13および14を含むクロック発生部を示す。

【0020】この情報処理装置1は、1チップで構成することもできるし、システム・バス10を除く各構成要素を個別のチップで構成することもできる。また、一部

を1チップで、他の一部を個別のチップで構成することもできる。

【0021】システム・バス10には、ROM11およびRAM19のアドレスを伝達するためのアドレス・バスと、ROM11、RAM19、CPU17、またはCPU18のデータ（命令、処理結果等）を伝達するためのデータ・バスとが含まれている。このシステム・バス10には、ROM11、クロック制御装置12、RAM19、ならびにCPU17および18が接続されている。

【0022】本実施の形態では、CPU17および18に共通のプログラム・カウンタ15が設けられている。このため、CPU17および18は、通常のCPUと同様に、図示しない論理演算装置（ALU）、レジスタ群等を備えているが、プログラム・カウンタを備えていない。

【0023】RAM19は、CPU17および18のワーク・エリアとして使用される。

【0024】ROM11には、VLIW（Very Long Instruction Word）で記述されたプログラムが記憶されている。図2は、VLIWのアセンブリ言語で記述されたプログラムの一例を示し、同図（A）は、本実施の形態に係るプログラムを、同図（B）は、参考として従来のプログラムを、それぞれ示している。

【0025】図2（A）に示すプログラムは、2つのCPUによる処理を前提に記述されたものである。このプログラムは、高級言語（たとえばC言語等）で記述したプログラムをVLIW用のコンパイラを用いてコンパイルすることにより生成することもできるし、ユーザがアセンブリ言語を用いて直接記述することもできる。

【0026】VLIWで記述されたプログラムは、原則として、複数（本実施の形態では2つ）のCPUがそれぞれ実行する命令群を1セットとして記述される。ただし、後述するように、CPUへのクロック信号の供給の停止および開始を指示する命令は、本実施の形態では、CPU17または18が実行する命令ではなく、クロック制御装置12が実行する命令である。また、クロック信号の供給が停止されているCPUに対する命令は記述されないか、または、“NULL”という特殊なコードが記述されるようになっている。

【0027】たとえば、アドレス*i*の命令（ニーモニック・コード）“MOV A, B; C, D”は、CPU17が実行する命令“MOV A, B”（CPU17のレジスタAの内容をCPU17のレジスタBにコピーする）と、CPU18が実行する命令“MOV C, D”（CPU18のレジスタCの内容をCPU18のレジスタDにコピーする）とを含んでいる。

【0028】アドレス（ $i + \alpha$ ）の命令“STOP CPU1; ADD B, D, E”は、クロック制御装置12が実行する命令“STOP CPU1”（CPU17

へのクロック信号の供給を停止する）と、CPU18が実行する命令“ADD B, D, E”（CPU18のレジスタBの内容とレジスタDの内容とを加算し、加算結果をレジスタEに格納する）とを含んでいる。

【0029】アドレス（ $i + 2\alpha$ ）の命令“（空欄またはNULL）; MUL B, D, F”は、CPU17が実行する命令を含まず、CPU18が実行する命令“MUL B, D, F”（CPU18のレジスタBの内容とレジスタDの内容とを乗算し、乗算結果をレジスタFに格納する）のみを含んでいる。CPU17に対する命令が記述される部分は、CPU17へのクロック信号の供給が開始されるまで、空欄または“NULL”の状態にされる。

【0030】アドレス（ $i + n\alpha$ ）の命令“RET CPU1; MOV G, H”は、クロック制御装置12が実行する命令“RET CPU1”（CPU17へのクロック信号の供給を開始（再開）する）と、CPU18が実行する命令“MOV G, H”（CPU18のレジスタGの内容をCPU18のレジスタHにコピーする）とを含んでいる。

【0031】これらの各命令は、図2のアドレスが示すように、ROM11において、一定のアドレス間隔 α （ α は正の整数）ごとに記憶されている。すなわち、各命令のバイト数は α バイト以下であり、 α バイトに満たない命令については、残りの部分は空けられることとなる。この α バイトのうち、たとえば、上位アドレス側に、CPU17が実行する命令が、下位アドレス側に、CPU18が実行する命令が、それぞれ格納されている。

【0032】また、CPU17および18が同時に実行する命令の処理時間（命令読み出しから命令実行終了までの時間（クロック数））は、同じとなるように構成される。したがって、CPU17が処理時間の相対的に短い命令（たとえば加算ADD）を実行し、CPU18が相対的に長い命令（たとえば除算DIV）を実行する場合には、相対的に長い命令は相対的に短い命令と同じ時間で処理可能な命令に分解されることとなる。

【0033】なお、図2（A）に示すプログラムを従来のVLIWのプログラムで記述すると、同図（B）に示すように、命令“STOP CPU1”の部分と、空欄または“NULL”が記述される部分と、命令“RET CPU1”の部分とは、命令“NOP”が記述されるようになっている。したがって、従来では、CPU17は、クロック信号が供給され、命令“NOP”を実行していた。

【0034】図2（A）および（B）から明らかなように、従来のVLIWのプログラムにおいて、ある1つのCPUに対する命令に“NOP”命令が2つ以上連続している部分があると、その先頭の命令“NOP”を命令“STOP [CPU名]”に置換し、最後の命令“N

OP”を命令“RET [CPU名]”に置換することにより、本発明におけるVLIWのプログラムを得ることができる。

【0035】図1に戻って、プログラム・カウンタ15は、実行すべき命令が記憶されているROM11のアドレスであるプログラム・カウンタ（以下「PC」という。）を生成し、CPU17および18ならびにクロック制御装置12に与えるものである。

【0036】プログラム・カウンタ15には、ROM11に記憶されたプログラムの先頭アドレスがあらかじめ設定されている。したがって、プログラムの実行開始時には、プログラム・カウンタ15はこの先頭アドレスをPCとしてCPU17および18ならびにクロック制御装置12に与える。

【0037】CPU17および18は、プログラム・カウンタ15から与えられたPCが示すアドレスから $[PC + \alpha - 1]$ が示すアドレスまでの α バイトのメモリの内容（すなわち図2（A）の1行分の命令）をシステム・バス10を介してROM11から読み出す。

【0038】CPU17は、読み出した α バイトの命令のうち、上位アドレス側にある命令（図2（A）における左側の命令）を実行し、CPU18は、下位アドレス側にある命令（図2（A）における右側の命令）を実行するようにあらかじめ設定されている。命令実行の完了と同時に、CPU17および18は、PCの更新を行う信号（以下「更新信号」という。）をセクタ16に出力する。

【0039】セクタ16は、CPU17および18からの更新信号のうちのいずれか一方を選択してプログラム・カウンタ15に出力する。セクタ16がCPU17および18からの更新信号のいずれを選択し、出力するかは、後述するようにクロック制御装置12により制御される。

【0040】プログラム・カウンタ15は、更新信号が与えられると、PCを α だけ増加させるように構成されている。したがって、プログラム・カウンタ15は、更新信号を受けることにより、PCを α だけ増加させ、これを新たなPCとしてCPU17および18ならびにクロック制御装置12に与える。その後、CPU17および18は、前述したROM11からの命令の読み出しおよび実行を行い、このような処理がプログラムが完了するまで繰り返される。

【0041】クロック制御装置12は、ROM11に記憶されたプログラムに従って、クロック発生器13および14の動作を制御するとともに、セクタ16の信号の選択を制御するものである。

【0042】また、クロック発生器13および14は、ともにクロック信号を生成し、CPU17および18にクロック信号をそれぞれ供給するものである。クロック発生器13および14は、クロック信号を供給するCPU

Uの特性に応じて、ともに同じ周期のクロック信号を生成することもできるし、異なる周期のクロック信号を生成することもできる。

【0043】前述したように、クロック制御装置12にも、プログラム・カウンタ15からPCが与えられる。クロック制御装置12は、PCが示すアドレスから $[PC + \alpha - 1]$ が示すアドレスまでの α バイトのメモリの内容（命令）をシステム・バス10を介してROM11から読み出す。

【0044】クロック制御装置12は、読み出した α バイトの命令に“STOP [CPU名]”または“RET [CPU名]”が含まれているかどうかを判断する。そして、クロック制御装置12は、“STOP [CPU名]”が含まれている場合には、[CPU名]に対応するCPUへのクロック信号の供給を停止し、“RET [CPU名]”が含まれている場合には、[CPU名]に対応するCPUへのクロック信号の供給を開始（再開）し、いずれも含まれていない場合には、これまでの状態を維持する。

【0045】すなわち、クロック制御装置12は、“STOP [CPU名]”が含まれている場合において、[CPU名]がCPU1のときは、クロック発生器13のクロック信号の供給を停止し、[CPU名]がCPU2のときは、クロック発生器14のクロック信号の供給を停止する。一方、“RET [CPU名]”が含まれている場合において、[CPU名]がCPU1のときは、クロック発生器13のクロック信号の供給を開始し、[CPU名]がCPU2のときは、クロック発生器14のクロック信号の供給を開始する。

【0046】クロック信号の供給の停止には、クロック発生器12および13の稼働を停止させる方法と、クロック発生器12および13の稼働を継続した状態で、クロック信号がこれらのクロック発生器12および13から出力されることを停止する方法とが含まれる。クロック信号の供給の開始（再開）についても、稼働を停止したクロック発生器12および13の稼働を開始（再開）させる方法と、稼働継続中のクロック発生器12および13からのクロック信号の出力を開始（再開）する方法とが含まれる。稼働を停止したクロック発生器の稼働を開始（再開）させる方法を採用した場合には、稼働開始（再開）時に、稼働を開始するクロック発生器のクロック信号と、もう一方の稼働継続中のクロック発生器のクロック信号とのクロック波形の同期（立ち上がりエッジおよび立ち下がりエッジの一致）が行われるようになっている。

【0047】なお、クロック制御装置12がクロック発生器13および14のいずれを制御するかは、[CPU名]により判断する以外に、“STOP”命令または“RET”命令の位置により判断することもできる。すなわち、クロック制御装置12は、これらの命令がRO

M11から読み出した命令の上位アドレス側にある場合には、クロック発生器13を、下位アドレス側にある場合には、クロック発生器14を、それぞれ制御することもできる。

【0048】クロック制御装置12は、クロック信号が供給され稼動しているCPUからの更新信号がプログラム・カウンタ15に与えられるようにセクタ16の制御も行う。すなわち、クロック制御装置12は、クロック発生器13のクロック信号の供給を停止した場合には、CPU18からの更新信号がプログラム・カウンタ15に与えられるようにセクタ16を制御し、一方、クロック発生器14のクロック信号の供給を停止した場合には、CPU18からの更新信号がプログラム・カウンタ15に与えられるセクタ16を制御する。CPU17および18の双方にクロック信号が供給され、双方が稼動している場合には、いずれか一方の任意の更新信号がプログラム・カウンタ15に与えられるようにセクタ16を制御する。

【0049】図3は、図2(A)に示すプログラムを情報処理システム1で実行した場合における、クロック発生器13のクロック信号CLK1およびクロック発生器14のクロック信号CLK2の波形を示すタイム・チャートである。命令“STOP CPU1; ADD B, D, E”の実行により、クロック発生器13のクロック信号CLK1の供給が停止し、命令“RET CPU1; MOV G, H”の実行により、クロック発生器13のクロック信号CLK1の供給が開始(再開)される様子が示されている。

【0050】このように、本実施の形態によると、従来ではクロック信号の供給を受けて“NOP”命令を実行するCPUへのクロック信号の供給が停止される。クロック信号の供給が停止されることにより、そのCPUによる消費電力を節約することができる。また、消費電力の節約は、発熱の減少をもたらし、その結果、熱を外部に排出するためのファン等のコスト、ファンによって消費される電力の軽減にも寄与する。

【0051】さらに、CPU17および18は共通のPCに従って処理を実行するので、プログラムのデバッグまたは情報処理システムを構成するハードウェアのデバッグが行い易くなる。

【0052】<第2の実施の形態>図4は、本発明の第2の実施の形態に係る情報処理システム2の構成を示すブロック図である。この情報処理システム2は、システム・バス10、ROM11、クロック制御装置12、クロック発生器13および14、プログラム・カウンタ15、セクタ16、RAM19、ならびに「処理装置」の一例としてのCPU17および18を備えている。この情報処理システム2において、第1の実施の形態に係る情報処理システム1と同じ構成要素には同じ符号を付し、その説明を省略することとする。

【0053】この情報処理装置2は、1チップで構成することもできるし、システム・バス10を除く各構成要素を個別のチップで構成することもできる。また、一部を1チップで、他の一部を個別のチップで構成することもできる。

【0054】情報処理システム2が情報処理システム1と異なる部分は、クロック制御装置12の代わりにクロック制御装置22を備えている点である。このクロック制御装置22は、クロック発生器13および14の制御に加えて、プログラム・カウンタ15に更新信号を送出する点で、クロック制御装置12と異なる。この点については、後に詳述する。

【0055】第2の実施の形態においては、ROM11に、シングル・スレッド型のプログラムが記憶される。シングル・スレッド型のプログラムとは、複数(本実施の形態では2つ)のCPUが1つのプログラムを立ち代り交代で実行するように作成されたプログラムをいう。したがって、任意の時刻において処理を実行しているCPUは、常に1つである。たとえば、演算処理を高速に実行できるCPUとデータ転送処理を高速に実行できるCPUとがあった場合に、1つのプログラムにおける演算処理命令の部分は前者のCPUに実行させ、データ転送処理命令の部分は後者のCPUに実行させるように、プログラムは作成される。このように、CPUの特性に応じて、処理を機能分散することにより、プログラムをより高速に実行することができる。

【0056】図5は、アセンブリ言語で記述されたシングル・スレッド型のプログラムの一例を示している。このプログラムは、2つのCPUによる処理を前提に記述されたものである。また、このプログラムは、高級言語(たとえばC言語等)で記述したプログラムをシングル・スレッド用のコンパイラを用いてコンパイルすることにより生成することもできるし、ユーザがアセンブリ言語を用いて直接記述することもできる。

【0057】シングル・スレッド型のプログラムは、通常のアセンブリ言語と同様に、シーケンシャルな命令列で記述されるが、通常の命令に加えて、命令を実行するCPUを交代し、かつ、クロック信号の供給停止/開始を行うための命令を含んでいる。本実施の形態では、通常の命令は、CPU17または18が実行し、命令を実行するCPUを交代し、かつ、クロック信号の供給停止/開始を行うための命令は、クロック制御装置22が実行する。

【0058】たとえば、アドレス($i+2\beta$)の命令“JMP CPU2”は、命令を実行するCPUをCPU17からCPU18へ交代するとともに、“CPU1”へのクロック信号の供給を停止し、“CPU2”へのクロック信号の供給を開始(再開)するものである。また、アドレス($i+n\beta$)の命令“JMP CPU1”は、この逆の処理を行うものである。

【0059】他のアドレス i の命令“MOV A, B”, アドレス $(i+\beta)$ の命令“MUL B, C, D”, アドレス $(i+3\beta)$ の命令“MOV D, E”, およびアドレス $(i+4\beta)$ の命令“MOV F, G”は、前述した第1の実施の形態のものと同様である。

【0060】これらの各命令は、ROM11において、一定のアドレス間隔 β (β は正の整数)ごとに記憶されていてもよいし、各命令のバイト数に応じた間隔ごとに記憶されていてもよい。前者の場合に、プログラム・カウンタ15は、更新信号が与えられるごとに、PCを β ずつ増加させるように設定される。一方、後者の場合に、CPU17もしくは18またはクロック制御装置22は、PCの増分を含む更新信号をプログラム・カウンタ15に与え、プログラム・カウンタ15は、与えられた増分ずつPCを増加させるように設定される。

【0061】プログラムの実行開始時において、CPU17または18のいずれが実行を開始するかは、情報処理システム2に初期設定としてあらかじめ設定しておくこともできるし、プログラムの先頭位置に命令“JMP CPU1”または命令“JMP CPU2”を含めることにより指定することもできる。これにより、プログラムの実行開始時には、クロック発生器13または14のいずれか1つからクロック信号が供給され、CPU17または18の一方のみが稼働状態に置かれる。

【0062】クロック制御装置12は、CPU17または18と同様に、プログラム・カウンタ15から与えられたPCにより、ROM11から各命令を読み出す。そして、クロック制御装置12は、読み出した命令がアドレス $(i+2\beta)$ の命令“JMP CPU2”である場合には、クロック発生器13からCPU17へのクロック信号の供給を停止し、クロック発生器14からCPU18へのクロック信号の供給を開始(再開)する。またこの処理とともに、クロック制御装置12は、CPU18からの更新信号がプログラム・カウンタ15に与えられるようにセレクタ16を制御するとともに、プログラム・カウンタ15に更新信号を与える。

【0063】これにより、CPU18は稼働を開始(再開)する。また、プログラム・カウンタ15はPCをアドレス $(i+3\beta)$ に更新する。その後、CPU18は、アドレス $(i+3\beta)$ 以降の命令を逐次実行して行く。

【0064】アドレス $(i+n\beta)$ では、アドレス $(i+2\beta)$ とは逆の処理が行われ、CPU17が稼働を開始(再開)する。

【0065】なお、参考までに、従来の情報処理システムにおいてシングル・スレッド型のプログラムを実行する場合には、CPU17が処理を実行中、CPU18は待機状態に置かれ、クロック信号がCPU18に供給され続ける。また、CPU18が処理を実行中、CPU1

7は待機状態に置かれ、クロック信号がCPU17に供給され続ける。

【0066】図6は、図5に示すプログラムを情報処理システム2で実行した場合における、クロック発生器13のクロック信号CLK1およびクロック発生器14のクロック信号CLK2の波形を示すタイム・チャートである。命令“JMP CPU2”の実行により、クロック発生器13のクロック信号CLK1の供給が停止されるとともに、クロック発生器14のクロック信号CLK2の供給が開始(再開)される様子が示されている。また、命令“JMP CPU1”の実行により、クロック発生器13のクロック信号CLK1の供給が開始(再開)されるとともに、クロック発生器14のクロック信号CLK2の供給が停止される様子が示されている。

【0067】このように、本実施の形態によると、従来ではクロック信号の供給を受け、待機状態に置かれるCPUへのクロック信号の供給が停止される。クロック信号の供給が停止されることにより、そのCPUによる消費電力を節約することができる。また、消費電力の節約は、発熱の減少をもたらし、その結果、熱を外部に排出するためのファン等のコスト、ファンによって消費される電力の軽減にも寄与する。

【0068】さらに、CPU17および18は共通のPCに従って処理を実行するので、プログラムのデバッグまたは情報処理システムを構成するハードウェアのデバッグが行い易くなる。

【0069】<他の実施の形態>前述した第1の実施の形態および第2の実施の形態において、処理を実行しないCPUに対して、クロック信号の供給を停止せずに、クロック発生器13および14が有する分周器の分周率を変化させて、クロック信号の周期を、命令の実行時よりも長く(たとえば2倍、4倍等に)することもできる。また、クロック信号の周期を長くされたCPUが処理を再開する場合には、クロック信号の周期を元に戻すこともできる。これによっても消費電力の軽減を図ることができる。

【0070】また、第1の実施の形態および第2の実施の形態において、クロック発生部3を、図7に示すように構成することができる。このクロック発生部3は、クロック制御装置32、クロック発生器13、ならびにゲート30および31を備えている。

【0071】クロック制御装置32は、クロック制御装置12(図1および図4参照)のようにクロック発生器13および14(図1および図4参照)を直接制御するのではなく、ゲート31および32の開閉を制御する。

【0072】ゲート31および32は、開いている状態では、クロック発生器13からのクロック信号を通過させて、CPU17および18へクロック信号をそれぞれ供給し、閉じている状態では、クロック発生器13からのクロック信号を遮断して、CPU17および18へク

ロック信号をそれぞれ供給しない。

【0073】クロック発生部3をこのように構成することによっても、クロック信号の供給の停止／開始を制御することができる。また、この場合に、クロック発生器は1つで済むこととなり、構成の簡単化およびコストの低減を図ることができる。さらに、クロック発生器は1つであるので、2つのクロック発生器を設けた場合における両クロック発生器のクロック信号の同期を図る必要もなくなる。

【0074】なお、これまで述べた実施の形態は、いずれも「処理装置」の一例としてのCPUが2つの場合を説明したが、3つ以上の場合であっても、本発明を適用することができるのはいうまでもない。また、CPUは「処理装置」の一例であって、「処理装置」は、デジタル信号処理プロセッサ(DSP)、浮動小数点処理装置等であってもよい。

【0075】<付記>

(付記1) クロック発生器と、該クロック発生器が生成するクロック信号がそれぞれ供給され、該クロック信号に同期してそれぞれに割り当てられた処理を実行する複数の処理装置とを備えている情報処理システムにおいて、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給するクロック制御装置を備えている、ことを特徴とする情報処理システム。

(付記2) 付記1において、前記クロック制御装置が、前記クロック信号の供給が停止された処理装置が処理を再開する場合には、該処理装置へのクロック信号の供給を開始し、前記処理の実行時の周期よりも長い周期のクロック信号を供給された処理装置が処理を再開する場合には、該処理装置へのクロック信号の周期を元に戻す、ことを特徴とする情報処理システム。

(付記3) 付記1において、プログラムを記憶するメモリと、前記メモリに記憶された前記プログラムを構成する命令のうち実行対象となる命令が記憶されたアドレスを順次示すプログラム・カウンタを生成して、前記複数の処理装置の少なくとも2つの処理装置に与えるプログラム・カウンタと、をさらに備え、前記少なくとも2つの処理装置が、前記メモリに記憶されたプログラムを前記プログラム・カウンタから与えられる共通のプログラム・カウンタに従ってそれぞれ実行する、ことを特徴とする情報処理システム。

(付記4) 付記3において、前記プログラムが、前記クロック信号の前記処理装置への供給を停止する停止命令と、前記停止命令の対象となる処理装置を指定する指定情報とを備え、前記クロック制御装置が、前記停止命令および前記指定情報に従って、前記指定情報により指定された処理装置へのクロック信号の供給を停止する、ことを特徴とする情報処理システム。

(付記5) 付記2において、プログラムを記憶するメモリと、前記メモリに記憶された前記プログラムを構成する命令のうち実行対象となる命令が記憶されたアドレスを順次示すプログラム・カウンタを生成して、前記複数の処理装置の少なくとも2つの処理装置に与えるプログラム・カウンタと、をさらに備え、前記少なくとも2つの処理装置が、前記メモリに記憶されたプログラムを前記プログラム・カウンタから与えられる共通のプログラム・カウンタに従ってそれぞれ実行し、前記プログラムが、前記クロック信号の前記処理装置への供給を開始する開始命令と、前記開始命令の対象となる処理装置を指定する指定情報とを備え、前記クロック制御装置が、前記開始命令および前記指定情報に従って、前記指定情報により指定された処理装置へのクロック信号の供給を開始する、ことを特徴とする情報処理システム。

(付記6) 付記3から5のいずれか1つにおいて、前記少なくとも2つの処理装置が、前記プログラムのうち同時並行に実行できる部分を同時並行して実行する、ことを特徴とする情報処理システム。

(付記7) 付記6において、前記プログラムがVLIW型のプログラミング言語で記述されている、ことを特徴とする情報処理システム。

(付記8) 付記3から5のいずれか1つにおいて、前記少なくとも2つの処理装置が、該プログラムを立ち代り交代で実行する、ことを特徴とする情報処理システム。

(付記9) 付記1から8のいずれか1つにおいて、前記クロック制御装置が、前記クロック発生器の動作を停止することにより、前記クロック信号の供給を停止するものである、ことを特徴とする情報処理システム。

(付記10) 付記1から8のいずれか1つにおいて、前記クロック制御装置が、前記クロック発生器の動作を継続させた状態で、前記クロック発生器からのクロック信号の出力を停止することにより、前記クロック信号の供給を停止するものである、ことを特徴とする情報処理システム。

(付記11) 付記1から8のいずれか1つにおいて、前記クロック発生器と前記処理装置との間に配置され、前記クロック発生器からのクロック信号の通過／遮断を行うゲートとをさらに備え、前記クロック制御装置が、前記ゲートの通過／遮断を制御するものである、ことを特徴とする情報処理システム。

(付記12) 付記1から11のいずれか1つにおいて、前記クロック制御装置が、前記クロック発生器が有する分周器の分周率を変化させることにより、その周期を処理の実行時の周期よりも長くするものである、ことを特徴とする情報処理システム。

(付記13) クロック信号の供給を受け、該クロック信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御装置であって、

前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給するようにクロック信号の供給を制御する、クロック制御装置。

(付記14) 付記13において、前記クロック信号の供給が停止された処理装置が処理を再開する場合には、該処理装置へのクロック信号の供給を開始し、前記処理の実行時の周期よりも長い周期のクロック信号を供給された処理装置が処理を再開する場合には、該処理装置へのクロック信号の周期を元に戻すようにクロック信号の供給を制御する、クロック制御装置。

(付記15) クロック信号の供給を受け、該クロック信号に同期して処理を実行する複数の処理装置へのクロック信号の供給を制御するクロック制御方法であって、前記複数の処理装置のうち、処理を実行しない処理装置に対し、前記クロック信号の供給を停止し、または、処理の実行時の周期よりも長い周期の前記クロック信号を供給する、クロック制御方法。

(付記16) 付記15において、前記クロック信号の供給が停止された処理装置が処理を再開する場合には、該処理装置へのクロック信号の供給を開始し、前記処理の実行時の周期よりも長い周期のクロック信号を供給された処理装置が処理を再開する場合には、該処理装置へのクロック信号の周期を元に戻す、クロック制御方法。

【0076】

【発明の効果】本発明によると、クロック信号の供給が停止された処理装置の電力消費が軽減されるので、情報処理システム全体の消費電力が軽減される。また、消費電力の軽減は、発熱量の軽減にもつながり、発熱の増大、および、この熱を外部に放出するためのファン等による電力消費の増大も防止できる。

【0077】また、消費電力の軽減とともに、プログラ

ムのデバッグおよび情報処理システムを構成するハードウェアのデバッグが容易となる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態に係る情報処理システムの構成を示すブロック図である。

【図2】VLIWのアセンブリ言語で記述されたプログラムの一例を示し、(A)は、本実施の形態に係るプログラムを、(B)は、参考として従来のプログラムを、それぞれ示す。

【図3】図2(A)に示すプログラムを情報処理システムで実行した場合の、クロック発生部のクロック信号CLK1およびCLK2の波形を示すタイム・チャートである。

【図4】本発明の第2の実施の形態に係る情報処理システムの構成を示すブロック図である。

【図5】アセンブリ言語で記述されたシングル・スレッド型のプログラムの一例を示している。

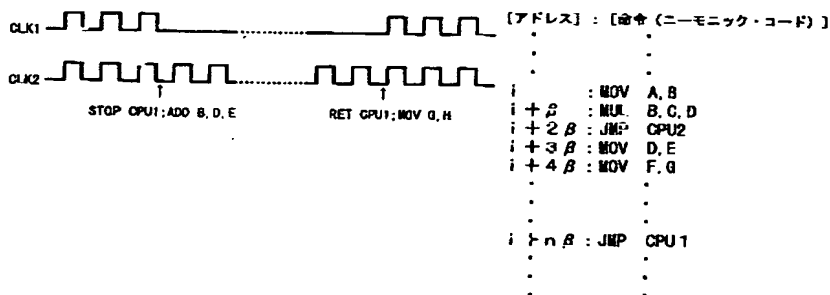
【図6】図5に示すプログラムを情報処理システムで実行した場合における、クロック発生部のクロック信号CLK1およびCLK2の波形を示すタイム・チャートである。

【図7】クロック発生部の他の実施の形態を示すブロック図である。

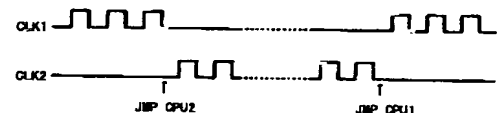
【符号の説明】

- 1, 2 情報処理システム
- 3 クロック発生部
- 11 ROM
- 12, 22, 32 クロック制御装置
- 13, 14 クロック発生器
- 15 プログラム・カウンタ
- 17, 18 CPU
- 30, 31 ゲート

【図3】

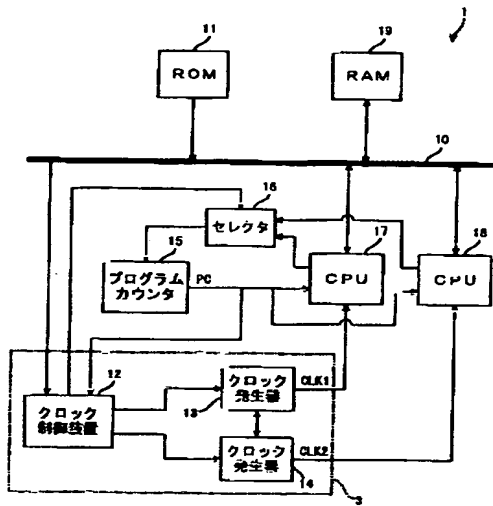


【図5】

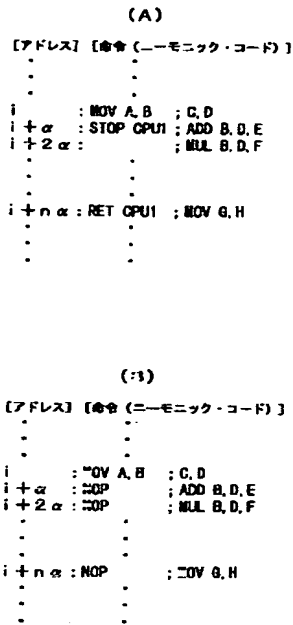


【図6】

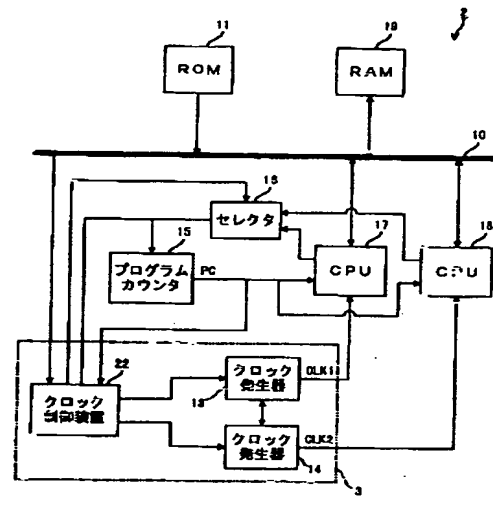
【図1】



【図2】



【図4】



【図7】

